Combining Heuristic Search, Visualization and Data Mining for Exploration of System Design Spaces

Martin S. Feather
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Dr
Pasadena CA 91109-8099
Martin.S.Feather@Jpl.Nasa.Gov

James D. Kiper   Selcuk Kalafat
Dept. of Computer Science and Systems Analysis
Miami University
Oxford, OH 45056
kiperjd@muohio.edu, kalafaas@muohio.edu

Abstract. Finding a near-optimal solution to a complex design problem is challenging. On the one hand the problem space is too large and convoluted for human comprehension, while on the other hand it is infeasible to elicit the entirety of design knowledge required for fully automatic problem solving.

We report on a practical application of information technology techniques to aid system engineers effectively explore large design spaces. We make use of heuristic search, visualization and data mining, the combination of which we have implemented within a risk management tool in use at JPL and NASA.

This approach is demonstrated on the planning for development of an advanced technology for spacecraft applications. In this context of risk-informed design, numerous risk abatement options give rise to a huge space of potential design solutions. We show how our approach enhances the system engineers' ability to explore this design space.

Introduction

Complexity and the role of information technology. Dealing with increasing complexity is a recurring challenge for systems engineering. Complexity stems from two sources: complexity within the design itself, and complexity of the space of possible designs. Complexity within the design itself (e.g., induced by subsystem interdependencies) makes understanding and evaluation of a design a challenging task; complexity of the design space (e.g., induced by options for design alternatives) makes selecting which design to adopt out of many possible designs a challenging task. Information technology aids system engineers in both these complexity arenas. Information technology allows engineering models of a design to be constructed and evaluated, so as to reveal properties of a candidate design ahead of its realization. Information technology allows models of the design space to be represented and explored, to reveal overall design tradeoff options, and to guide the identification of preferred design choices within that design space.

The effective use of information technology requires that it be blended with the expertise, insights and guidance of systems engineers. For many applications it is infeasible to elicit the entirety of design knowledge required for fully automatic problem solving. Instead, design must be a cooperative, iterative process between system engineers and the use of the information technology that supports them. An approach that supports such a blend is the "design by shopping" paradigm advocated in (Balling 1999). Information technology is used to model designs, to evaluate those models, and to visualize the results of those evaluations, so that system engineers can understand design tradeoffs and emerge with their preferred design.

Complexity in planning the development of spacecraft technologies. System engineers at JPL and NASA repeatedly face these same complexity challenges when planning the development of technologies for spacecraft applications. In this domain, design challenges stem from:

- Cross-disciplinary concerns (e.g., spacecraft involves navigation, propulsion, telecommunications). These concerns are cross-coupled and interact in multiple ways (e.g., electromagnetic interference, heat transfer).
- Severe constraints on the systems being developed and on the development process itself. Time and budget pressures constrain development; operational resources constrain the resulting system (e.g., mass, volume, power).
- Mission-critical issues. Spacecraft are critical systems that must operate correctly the first time in only partially understood environments, with almost no chance for repair.
- Unknowns: past experience provides only a partial guide when new mission concepts are to be enhanced and enabled by new technologies of which past experience is lacking.

Because of these challenging aspects of space missions, usually no one person has expertise that spans all the disciplines, or can simultaneously juggle all the factors involved in large and complex designs. Furthermore, much of the design skill is "tacit knowledge" in the heads of spacecraft experts, so it cannot be encoded in an automated tool. Therefore, key decision-making can be enhanced by a computer-aided, human-informed process.

In response to these challenges we have been pursuing a "design by shopping" like approach that emphasizes design that takes resource-effective risk-abatement into consideration in the trade space. Risk concerns play a prominent role in planning the design and development of spacecraft systems. Typically there are numerous decision alternatives (e.g., architecture choices, implementation alternatives) and risk abatement options (e.g., analyses, tests) with significant risk implications. Their costs (time, budget, etc.) and benefits (their effectiveness at risk abatement) have to be taken into account when selecting among them.

In this context the number of individual risk abatement options may be modest, but the choice of selections from among them is huge (e.g., given 50 binary options, there are $2^{50} \cong 10^{15}$ ways of selecting from among them). We use information technology to assist experts to make these selections. Specifically, we use a blend of modeling, heuristic search, data mining and visualization.

The main steps of our approach are sketched in Figure 1. An engineering design model is developed based on inputs from technical experts. Heuristic search is used to reveal the cost/benefit trade space implied by this model. System engineers identify the design neighborhood in which they are most interested. The space is culled to just the solutions within that neighborhood. System engineers provide "dissimilarity metrics" that indicate what they consider to be important distinctions among neighboring solutions. Dispersal and clustering algorithms are used to locate a modest number of distinct designs from within that neighborhood. Custom visualization presents the located designs to the system engineers, allowing them to make their choice of preferred design.

The remainder of the paper describes this approach in greater detail, and is structured as follows:

- We describe the key aspects of our risk-informed design methodology. We use this methodology to build early-lifecycle models of designs, which we use to evaluate the benefits we expect to attain from those designs, and the costs of developing those designs.

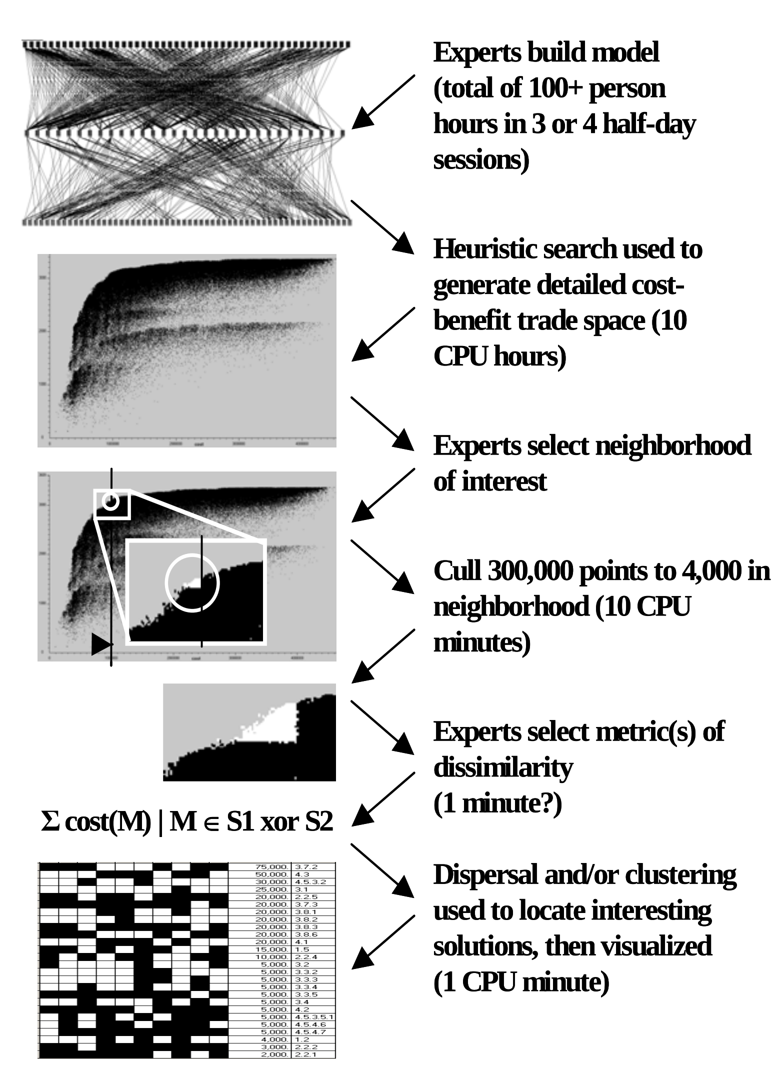


Figure 1. Overall approach.

- We show the use of heuristic search to locate (near) optimal designs, and visualization to reveal the overall cost-benefit trade space.
- We describe how system engineers, using the information revealed by heuristic search, select a "neighborhood of interest" in which they wish to concentrate their attention on locating a preferred design.
- We describe the innovative use of "dissimilarity metrics" to capture the system engineers' intuition of when designs are interestingly distinct. Our implementation uses these to extract from the previously identified neighborhood of interest a handful of interestingly distinct designs.
- We use visualization to further help the system engineers understand the identified designs.

Illustrative example. Throughout we use data from one of the technology assessment and infusion planning efforts performed at JPL. The details are proprietary, so we avoid revealing specifics. Nevertheless, all the quantities we report (e.g., the number of distinct mitigations) are actual figures from the assessment effort, and charts shown have all been generated from this data. Briefly, the assessment concerned an electronics packaging technique that has seen wide use on Earth, and on some space missions but only inside a temperature controlled housing. The focus of the assessment was its novel application to settings where the electronics would be exposed to the harsh conditions (e.g., extreme cold temperature) of planetary environments. The end goal was to identify and select appropriate design, fabrication, assembly and testing methods for the packaging technique so that it could be incorporated reliably into future spacecraft.

A Risk-Informed Design Methodology

A risk-informed design methodology underpins our work. Motivation for this stems from a vision of using risk as a *resource*, one that can be traded against other resources such as schedule, cost and performance (Greenfield, 1998). The methodology we have developed and applied at JPL and NASA supports this vision. It combines insights and skills of spacecraft experts, a model for representing their knowledge, a process for building and exploring the model, and custom software to support this process.

At its heart, it relies on users to identify: *objectives* to be achieved (and their relative priorities), the various *risks* to achieving those objectives, and options for risk *mitigation* (prevention, detection ahead of time, and alleviation). The connectivity among these pieces of information is as follows: risks are connected to the objectives that they would impact (should those risks occur), and mitigations are connected to the risks they reduce (should those mitigations be applied). Note that a risk may impact multiple objectives, an objective may be impacted by multiple risks, etc. Note also that different risk impacts and mitigation effects may have different strengths, for example, one risk may detract from one objective more than it does from another.

Models of actual technologies and systems are typically voluminous and convoluted, as illustrated by the data in Figure 2 extracted from the technology study used throughout this paper. This comprises 50 objectives, 31 risks, 58 mitigations, and some 800 links among them, numbers typical of the order of magnitude of data gathered in these assessment efforts.

This data was gathered from experts in a series of facilitator-led sessions, following the elicitation process we have established for our risk-centric models.

The risk-centric design model offers a selection of risk mitigations. For any given selection, the model can be evaluated to yield two measures:

- *cost*, calculated as the sum of the costs of the selected mitigations (e.g., the cost to perform a test), and of the repairs of risks they detect (e.g., the cost of fixing the bugs revealed by testing), and
- *benefit*, calculated as the sum total attainment of objectives taking risks into account. Risks detract from objectives' attainment, however risks themselves are reduced (in likelihood and/or impact) by the selected mitigations. Risk reduction leads to increased attainment of objectives.

For further details of our model, and how it is applied, see (Feather&Cornford 2003).

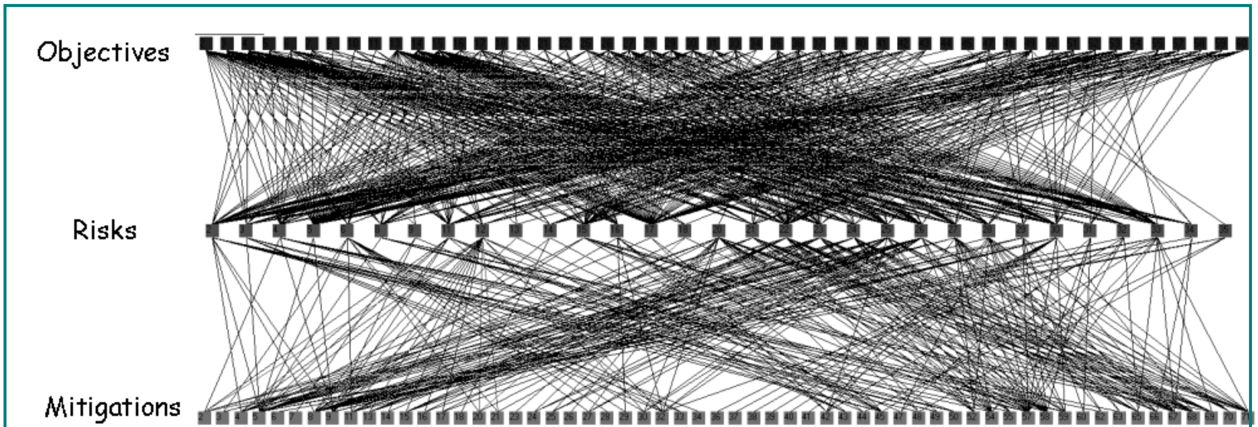


Figure 2. Topology of data in a completed risk model.

Optimal Designs and Heuristic Search

An optimal design is one that attains its objectives at minimal expense. Generally speaking mitigations increase expected objective attainment (by reducing risks), but incur costs. In most instances the total cost of all possible mitigations far exceeds the resources available. The primary purpose of our methodology is to help identify and select the set of mitigations to apply to achieve an optimal design within some cost bound.

In typical designs there will be many mitigations (dozens, possibly hundreds). The combinatorial choices from among these imply a space containing huge numbers of possible candidate solutions. In our application, 58 mitigations represented design and development choices whose costs range from the low thousands of dollars to, in a few cases, hundreds of thousands of dollars. Since there are 58 mitigations, there are in principle 2^{58} (approximately 10^{17}) different selections from among them.

We implemented simulated annealing (Kirkpatrick et al. 1983) within our risk toolset, and use it to locate near-optimal solutions. We have also explored other forms of heuristic search: genetic algorithms and machine learning – for a discussion of these, see (Cornford et al. 2003).

Results of simulated annealing. For our electronics technology dataset, a detailed simulated annealing search was performed, organized as a series of individual cost-bounded optimal searches at successive cost levels. The resulting cost-benefit trade space is shown in Figure 3. The sum total cost of all mitigations (approximately $4,750,000) determines the rightmost value of the x-axis, and the sum total value of all objectives (approximately 3,600) determines the topmost value of the y-axis. Its generation took on the order of 10 hours running on a 1.8 GHz PC.

Each of the approximately 300,000 individual points in the black "cloud" corresponds to a design solution (i.e., selection of mitigations). For a given solution, the software uses the quantitative risk-centric model to calculate cost and benefit. A small black point corresponding

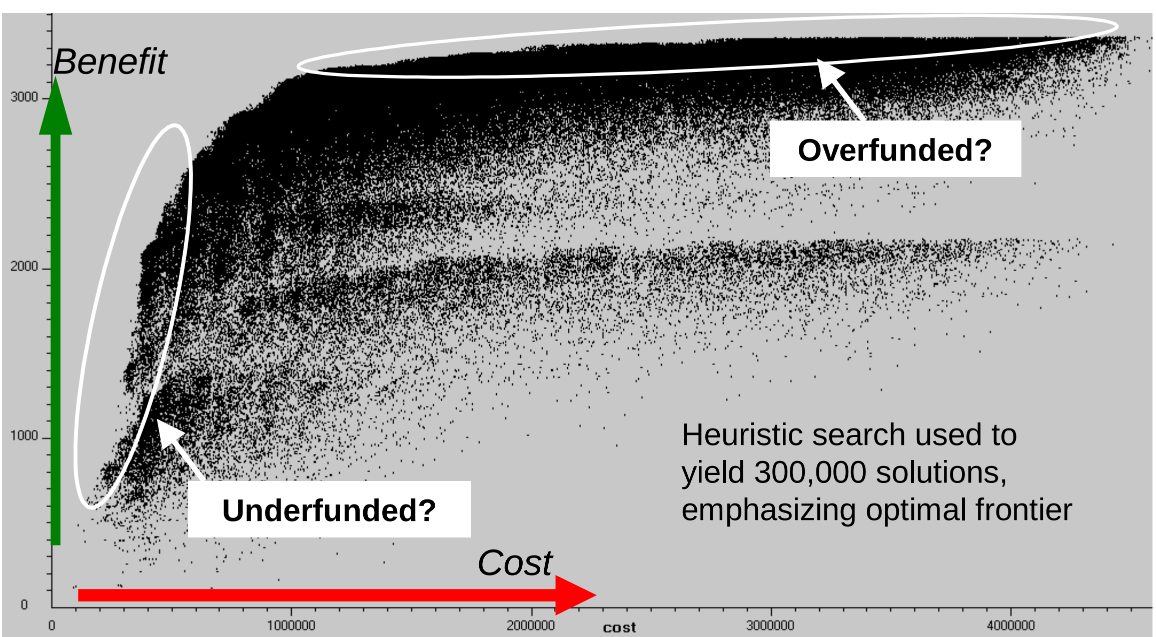


Figure 3. Cost-benefit trade space revealed by heuristic search

to the solution is then drawn on the plot – solution cost determines horizontal position, solution benefit vertical position. The upper-left frontier of the cloud is thus the “optimal” boundary, also referred to as the “Pareto front” (Sen&Yang 1998). Note that we plot a point for *every* solution investigated by the search, not just the “near-optimal” solution points on the boundary. This is important data, since the steps that follow investigate points close to, but not necessarily on, that optimal frontier. The simulated annealing search is designed to concentrate towards this optimal boundary.

The results of our simulated annealing search agree with the partial intuitions that system engineers had going into the technology study. At one extreme, they had contemplated funding the technology development at the $400,000 level, and had felt that to do so would severely limit what could be attained. At the other extreme, they had contemplated a large-scale infusion of funding ($3million or more), and had felt that this would be wasteful.

System Engineers Select Neighborhood of Interest

The system engineers know of funding availability, and level of benefit (attainment of objectives) desired. They use this knowledge to identify their neighborhood(s) of interest within the cost/benefit trade space revealed by the previous search.

The reasons we identify a *neighborhood* of interest, rather than simply picking specific near-optimal solutions on the frontier are twofold. First, the data over which the search is performed was produced through expert judgment. We do not assume all of those expert judgments to be perfectly accurate. That is, solutions within a small percentage of the near optimal solution may in truth be no more costly and/or attain no less benefit than ones calculated as the “near-optimal” in that neighborhood. Thus such a neighborhood encompasses different design solutions that, from the standpoint of cost and benefit, we judge to be equally acceptable within the accuracy of

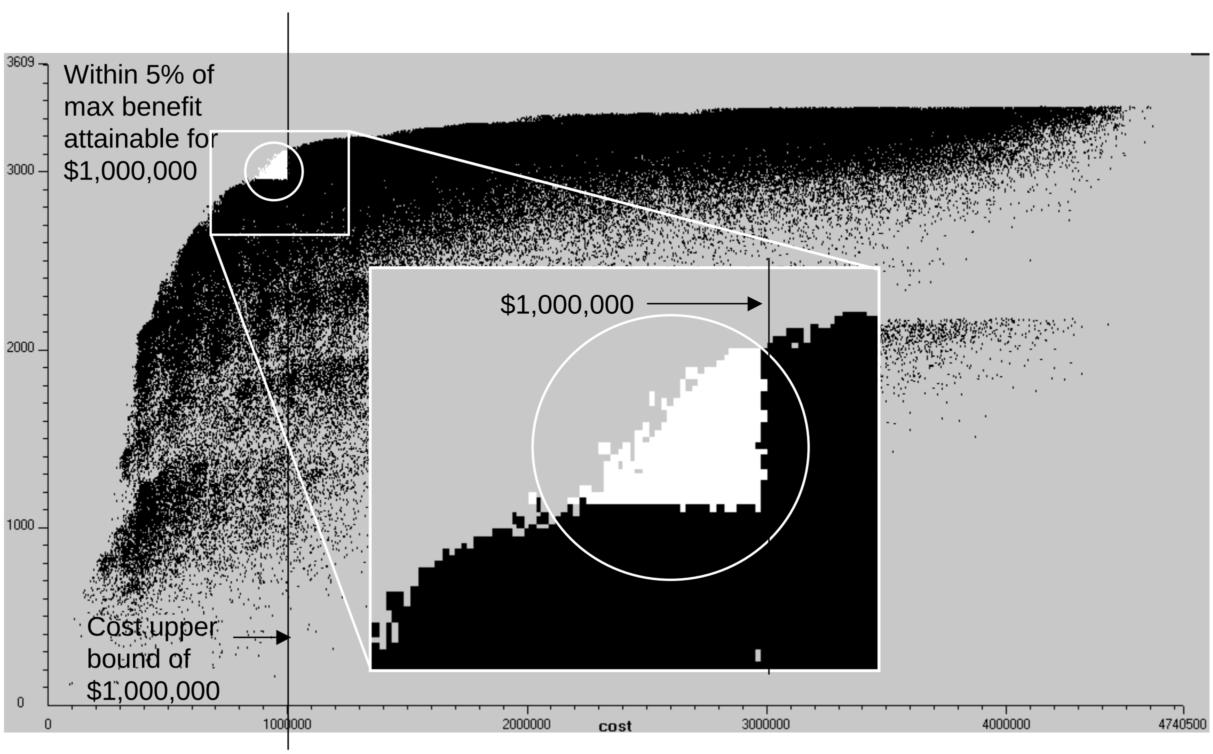


Figure 4. Neighborhood of interest.

our data. Second, there may be factors other than overall cost and benefit that would lead us to prefer one solution within this neighborhood over another. For example, one design solution in the neighborhood may make use of a hard-to-schedule test facility, while another does not, in which case we might prefer the latter.

While it should in principle be possible to encode such preferences within the utility function that guides the optimal search, we believe this to be infeasible in practice. In the first place, much of this may be "tacit" knowledge – not evident until the experts see concrete examples of design solutions. Even if the experts could foresee all these preferences in advance, our suspicion is that it would be a waste of time to ask them to try to articulate them all. Better, have them list just the ones that they recognize as the major impediments, perform the search, identify the neighborhood, scrutinize the results, and select the preferred solution.

Results of identifying a neighborhood of interest. An expert-identified neighborhood of interest defined as solutions costing ≤ $1,000,000 and attaining ≥ 95% of the objectives attainment of the best solution found at or below that cost limit is shown superimposed and magnified in Figure 4. For the dataset, 3,391 solutions (i.e., distinct selections of mitigations) fall within the neighborhood.

Dissimilarity Metrics

Manual scrutiny of each of 3,391 solutions is tedious at best. Since many will be very similar (differing by a small number of the selected mitigations), we use data mining to more effectively explore this neighborhood. We define a ***metric of dissimilarity*** – two design solutions (each a set of selected mitigations) will have a larger dissimilarity value according to this the more they differ from one another. Using this metric, solutions that lie within the neighborhood of interest, but which are distinct from one another with respect to this dissimilarity metric, are located and presented to the users.

In our risk-centric model, a design solution is described by a set of mitigations, each of which is either "on" or "off". A dissimilarity metric takes as input a pair of design solutions, and returns a numerical measure of dissimilarity.

A simple metric is to count the number of mitigations over which two design solutions differ, that is, count how many mitigations are "on" in one but not both of those solutions. If bit strings represent solutions, each bit corresponding to a mitigation, with value 1 if that mitigation is "on", 0 otherwise, then this metric is the *Hamming distance* between solutions' representations.

Another metric is to sum the costs of all the mitigations that are "on" in one but not both of two solutions. This metric therefore ranks as more dissimilar solutions that differ by higher cost mitigations.

Yet another metric makes use of relevant groupings of mitigations. For example, in our technology assessment studies, it is common to classify mitigations according to the phase in which they would apply – design, fabrication, assembly, test. We can define a metric of dissimilarity based on the difference in costs *between* these phases.

The experts using our software select which metrics to use to explore the neighborhood of design solutions.

Data Mining for Interesting Solutions

Given a neighborhood of solutions, and a dissimilarity metric chosen by the experts, we use two data mining techniques to locate solutions distinct with respect to the chosen metric:

- *Dispersal:* a set of maximally dispersed (using the metric of "dissimilar") solutions is extracted from the neighborhood of interest.
- *Clustering:* similar solutions, i.e., solutions that are close in distance according to the metric of "dissimilar", are aggregated into clusters.

Space limitations in this paper preclude the in-depth treatment of both techniques, so we will focus solely on the dispersal technique. Clustering is discussed briefly in (Menzies at al, 2003).

We have implemented a dispersal method that is a fast approximation of an idealized dispersal algorithm. It works as follows:

Input C (> 1) the number of dispersed solutions that the method is to find.
Let N be the set of design solutions in the neighborhood of interest:
Initialize S to be the singleton set holding the optimal design solution in N;
While S's cardinality < C, do:

- Find a design solution *ds* in (N – S) such that *ds*'s minimum distance from all the design solutions in S is as great as possible.
- Add *ds* to S.

Results of data mining using dispersal. Our approximate dispersal algorithm was applied to find 10 dispersed solutions with respect to each of the three metrics discussed above from the expert-identified neighborhood. It took under 2 minutes to find and plot the visualization of the 30 dispersed solutions – 10 dispersed solutions for each of the three metrics. Table 1 shows the dispersal distances between solution point s_1 and its shortest metric distance to the points $s_2 \ldots s_n$ where n = card(N)

	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
Hamming distance	23	20	18	18	18	17	17	16	16
Cost of different mitigations ($10,000's)	65	57	50	45	42	40	37	36	36
Cost difference of phases ($10,000's)	34	30	28	22	17	17	15	14	14

Table 1: Dispersal distances

Recall that our study involves a design with 58 distinct mitigation options. Observe that starting from the near-optimal solution that costs no more than $1million, our "hamming distance" metric led to the discovery of 9 further solutions within the 5% neighborhood that differed from the optimal by at least 16 and as many as 23 of those mitigation options (i.e., 27% to 39%). By way of contrast, the neighborhood's top 10 solutions (i.e., the 10 highest-benefit scoring solutions costing no more than $1million) differed from the optimal by as few as 2 and as many as 10 mitigation options (i.e., 3% to 17%).

The cost-based metrics led to similarly diverse solutions. The total cost of the mitigations by which the 9 further solutions each differ from the near optimal solution range from $360,000 to $650,000 (more details of these solutions are presented in the next section). By way of contrast, the neighborhood's top 10 solutions (i.e., the 10 highest-benefit scoring solutions costing no more than $1million) differed from the optimal by as little as $9,000 and as much as $129,000.

The diversity of the cost difference solutions shows a similar pattern – for the top 10 solutions, such diversity is modest, while for the dispersal-discovered solutions, significantly more diverse solutions are found.

Value of locating dispersed solutions. As we discussed earlier, the neighborhood of interest encompasses a large number of different design solutions that, from the standpoint of cost and benefit, we judge to be equally acceptable within the accuracy of our data. The value of our simple data mining technique lies in its ability to locate significantly diverse solutions with respect to a provided metric. By choosing a metric that corresponds to *interestingness* (the farther apart two solutions are, the more interestingly the difference between them), it locates solutions that as interestingly distinct as possible. Our results indeed exhibit this, as seen when we scrutinize the solutions in detail, considered next

Visualization of the Data Mining Results

Visualization is used to present results of the dispersal method of data mining to the system engineers, allowing them to see what distinct solutions are available, and to understand the ways in which those solutions differ (and the ways in which they do *not* differ).

We use a simple tabular display as the means to provide a visualization of the overall dispersal results. We also have available the risk tool's mechanisms for detailed scrutiny of individual solutions, and make particular use of capabilities to *compare* alternate solutions.

These are most easily explained with reference to examples taken from our application of spacecraft technology planning.

Tabular visualization of data mining results. Figure 5 shows our tabular display. Each of the rows represents a mitigation, listed with its cost, number and name (in order to protect the proprietary nature of this information, the portion of the table showing the mitigation names is omitted from this figure). Each of the columns of black and white squares represents a distinct solution (selection of mitigations). If a mitigation is included in a solution, the corresponding cell is filled in, otherwise it is left blank.

Since the focus is on how solutions differ, mitigations that are constant across all of the dispersal solutions (either always on or always off) are omitted. Hence every row in the displayed results involves at least one black square (mitigation included) and one white square (mitigation not included). The information about which mitigations remain

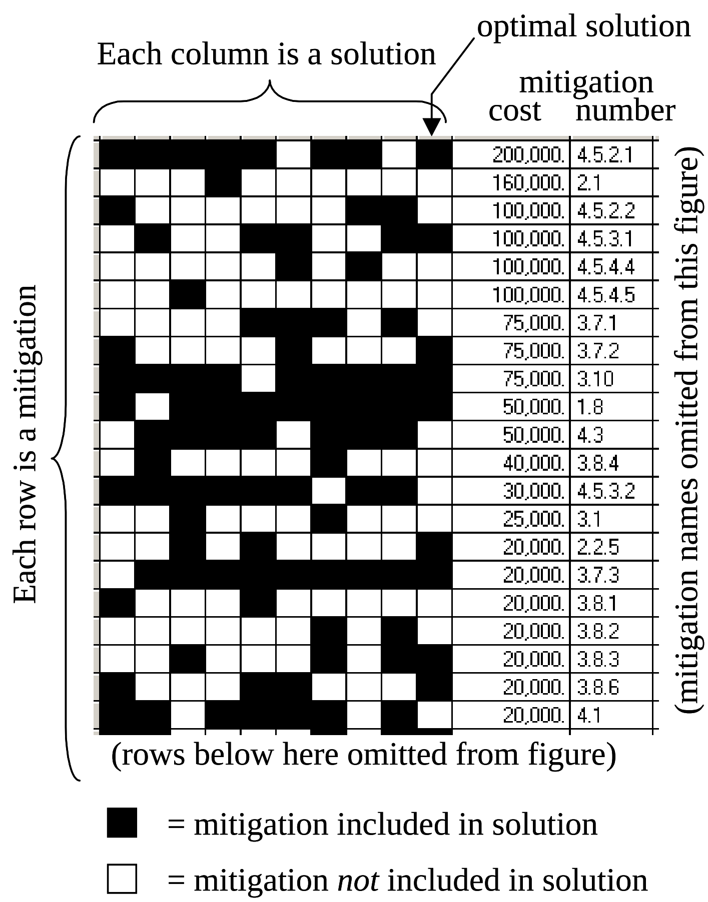


Figure 5. Tabular visualization of dispersal results.

consistently on (or off) over all the solutions is also of value, and is presented to the system engineers separately.

The display shows dispersal results using the metric of the sum of the costs of all the mitigations that are "on" in one but not both of two solutions. The mitigations are arranged in order of decreasing cost, to draw attention to the most expensive ones first.

The visualization reveals that within the neighborhood explored, there are several relatively high cost mitigations that play a role in some, but not all, of the solutions in that neighborhood. Recall that the neighborhood was constrained to solutions whose total cost is no more than $1million, and whose benefit is within 5% of the maximum benefit of solutions in that neighborhood. For example, note that there are six relatively high-cost ($100,000 and up) mitigations listed.

The system engineers, informed by this information, can contemplate whether any of the alternatives stand out as particularly appealing. Suppose, for example, that they know that mitigation number 4.5.3.1 (the fourth one down) will require access to a limited testing resource that is hard to obtain time on. Knowing this, they may prefer to avoid its use if possible. Several of the columns indicate solutions that do not rely on that particular mitigation.

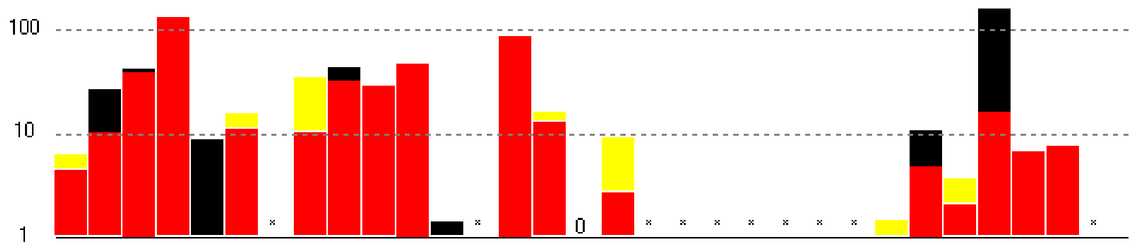
Another aspect of this visualization is that system engineers can see the details of each individual solution alongside one another. For example, five of the six top rows are mitigations that fall into the testing category (those mitigations whose number begins with the digit 4). From the table it is easy to see the available combinations from among these five relatively expensive options.

It is even possible that the system engineers will wish reject *all* these solutions, having now realized some constraint that was not previously articulated. In such an event, we would use that constraint as a filter on all the solutions originally discovered by search, yielding a revised cost-benefit trade space, and triggering reconsideration of the neighborhood of interest, etc.

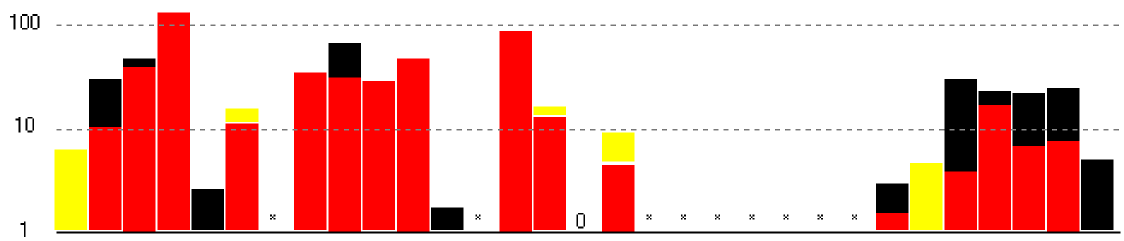
Comparison of individual solutions. In our risk-averse setting, system engineers wish to scrutinize the detailed ramifications of solution alternatives in terms of changes to individual risks, and changes to attainment of individual objectives. The tabular visualization above shows only which mitigations are involved. To understand the detailed ramifications we turn to use of our risk tool's bar chart display.

Figure 6 shows the bar chart display in use to compare changes to individual risks between solution alternatives. Our technology study involved 31 risks. The bar chart uses a vertical bar for each such risk. Each risk's magnitude (after taking into account the risk-reducing effects of the selected mitigations) is indicated by the heights of the bar (using a *log* scale on the vertical axis). We can obviously use such a bar chart to display the risk magnitudes of an individual solution. The figure shows comparison between two solutions. One solution has been adopted as the "baseline" (in the two examples shown, we set the "optimal" solution as this baseline), and an alternate solution is compared against it. Increase in risk in going from the optimal to the alternate is colored black, and decrease is colored red.

In the upper chart we can see at a glance a particularly striking increase in one particular risk – the one topped by the black bar that extends above the "100" level. Conversely, in the lower chart we see that the risk increase is more dispersed over several risks. The utility of information such as this is that it can focus attention on the (relatively few) places where solution alternatives have significantly different ramifications. The system engineers might, for example, be motivated to brainstorm on novel ways to mitigate that strikingly increased risk of alternative 1.



Risk increase (black) / decrease (yellow) in switching from optimal to alternative 1



Risk increase (black) / decrease (yellow) in switching from optimal to alternative 2

Figure 6. Comparison of alternative solution's individual risks.

We can also use the same charting capabilities to show ramifications in terms of objective attainment. In our technology studies, system engineers can utilize such information to guide them to locate the applications for which the technology is best suited.

Conclusions

Approach and related work. This paper has described an approach that applies information technology to assist system engineers explore large design spaces. The approach blends the computational power that information technology offers with the expertise, insights and guidance of system engineers. Appropriate visualizations are used to convey to the system engineers the salient aspects of the computational results. Its key elements are the following:

- Heuristic search, in the form of simulated annealing, has long been used for design optimization. The risk-centric model we use to evaluate points in the design space is somewhat novel. Its closest mainstream equivalent is Quality Function Deployment (QFD) (Akao 1999). Our model is specialized to risk concerns, and adopts a probabilistic interpretation of risk that is suited to the quantitative evaluations necessary in order to employ automated search. For information on our risk-informed methodology, "Defect Detection and Prevention (DDP)", and the software tool we have built to support it, see (Feather & Cornford, 2003); further details can be found at http://ddptool.jpl.nasa.gov
- Our use of dissimilarity metrics bears a resemblance to the "unexpectedness" measures of (Padmanabhan & Tuzhilin 1999), but is simpler to conduct because we seek only to locate unexpected data, not patterns in that data. We use some straightforward visualizations to convey the interesting design alternatives located using these metrics.
- Our overall approach matches the "design by shopping" paradigm (Balling 1999). An impressive blend of computation and visualization (offering user-driven selection of how to display data using a mixture of 3-D spatial location and color) that also matches this

paradigm is to be found in (Stump et al 2002). Our use of dissimilarity-metrics-based data mining substitutes for such sophisticated visualization capabilities. The other extreme, *more* reliance on computational reasoning over the design information, is exemplified by the approaches of a colleague of ours (Menzies and Hu 2003).

All the capabilities reported herein have been implemented within our DDP software for supporting risk-informed design.

Application. In this paper we have reported on a substantive practical application of these techniques to the selection of risk abatement solutions in the design of advanced technology. A study was performed at JPL to plan the development of an electronics packing technology for future spacecraft missions. While technologies for spacecraft use represent seemingly esoteric problems, the design challenges that arise in planning their development are widespread – cross-disciplinary concerns, resource constrained systems, risk averseness, and novel aspects of use. The complexity stemming from these challenges is also widespread

Future work. In future work we plan to investigate a closer connection between the dispersal and clustering techniques. In particular, we plan to use the former to rapidly get an overall feel for how solutions are dispersed, and use that information to guide slower but more revealing clustering algorithms. We also would like to take into account knowledge of uncertainty distributions in the input data to help better identify neighborhoods of interest.

Acknowledgements

The research described in this paper was done at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. We thank Steve Cornford, who leads the DDP effort at JPL, Ken Hicks and Ken Johnson for their guidance of technology studies using DDP, and Chet Borden for insightful comments on this presentation.

References

Akao, Y. 1990 "*Quality Function Deployment*", Productivity Press, Cambridge, Massachusetts.

Balling, R., "Design by Shopping: A New Paradigm?", *Proceedings of the Third World Congress of Structural and Multidisciplinary Optimization* (WCSMO-3) (Buffalo, NY, 1999), Vol. 1, pp 295-297.

Cornford, S.L., Feather, M.S., Dunphy, J.R., Salcedo, J. and Menzies, T., "Optimizing Spacecraft Design – Optimization Engine Development: Progress and Plans", *Proceedings of the 2003 IEEE Aerospace Conference* (Big Sky, Montana, March 2003), pp 7-3361 – 7-3368.

Feather, M.S. and Cornford, S.L., "Quantitative risk-based requirements reasoning", *Requirements Engineering* (Springer), Vol 8, No 4, 2003 pp. 248-265 – published online 25 February 2003, DOI 10.1007/s00766-002-0160-y.

Greenfield, M.A., "Risk Management - Risk As A Resource". Presentation to the Langley Research Center, May 14, 1998. http://www.hq.nasa.gov/office/codeq/risk/risk.pdf

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., "Optimization by simulated annealing". *Science, Number 4598, 13 May 1983*, 220, 4598:671–680.

Menzies, T., Kiper, J. and Feather, M.S., "Improved software engineering decision support through automatic argument reduction tools", *Proceedings of the 15th International Conference on Software Engineering and Knowledge Engineering* (San Francisco Bay, CA, July 2003).

Menzies, T. and Hu, Y., "Data Mining for Very Busy People" IEEE Computing, Nov. 2003, pp 18-25.

Padmanabhan, B. and Tuzhilin, A., "Unexpectedness as a measure of interestingness in knowledge discovery." *Decision Support Systems Journal*, 27, 3, 303-318, 1999.

Sen, P. and Yang, J-B., *Multiple Criteria Decision Support in Engineering Design*, Springer-Verlag, 1998.

Stump, G.M., Yukish, M. Simpson, T.W. and Bennet, L., "Multidimensional Visualization and Its Application to a Design by Shopping Paradigm", *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (Atlanta, GA, 2002)

Biography

Dr. Martin S. Feather is a Principal in the Software Quality Assurance Group at the Jet Propulsion Laboratory, California Institute of Technology. He works on developing research ideas and maturing them into practice, with current activities in the areas of software validation (analysis, test automation, V&V techniques) and of early phase requirements engineering and risk management. He works on the Defect Detection and Prevention effort (led by Dr. Steve Cornford – see http://ddptool.jpl.nasa.gov), and has been the primary architect and developer of DDP's software. For more details, see http://eis.jpl.nasa.gov/~mfeather

Dr. James D. Kiper is a Professor in the Department of Computer Science and Systems Analysis at Miami University. He teaches courses at a variety of levels in the CS curriculum. His research is in the general area of Software Engineering. In recent years, his research has centered on activities in the area of requirements engineering, software risk assessment, clustering and visualization. He has worked with Dr. Feather in refining some of the tools used in the DDP process.

Selcuk Kalafat is a graduate student in the Department of Computer Science and Systems Analysis at Miami University where he is pursuing a Master's degree. Over the past year and a half he has been working with Dr. Kiper in studying various search algorithms (simulated annealing, genetic algorithms), and clustering algorithms as applied to the DDP domain.